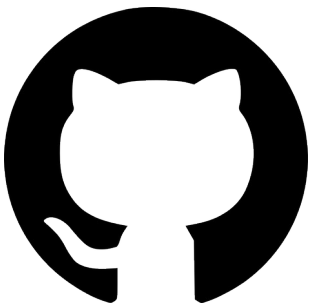
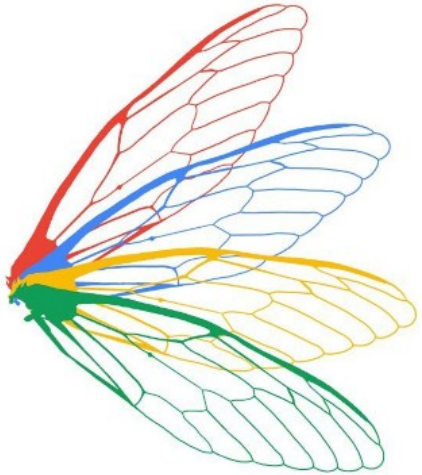
Git ve GitHub'a Giriş



Efe ÇİFTCİ, Ph.D.
Çankaya Üniversitesi Bilgisayar Mühendisliği Bölümü
Ekim 2023

efeciftci.com
academic/~efeciftci
in/efeciftci
efeciftci
efeciftci

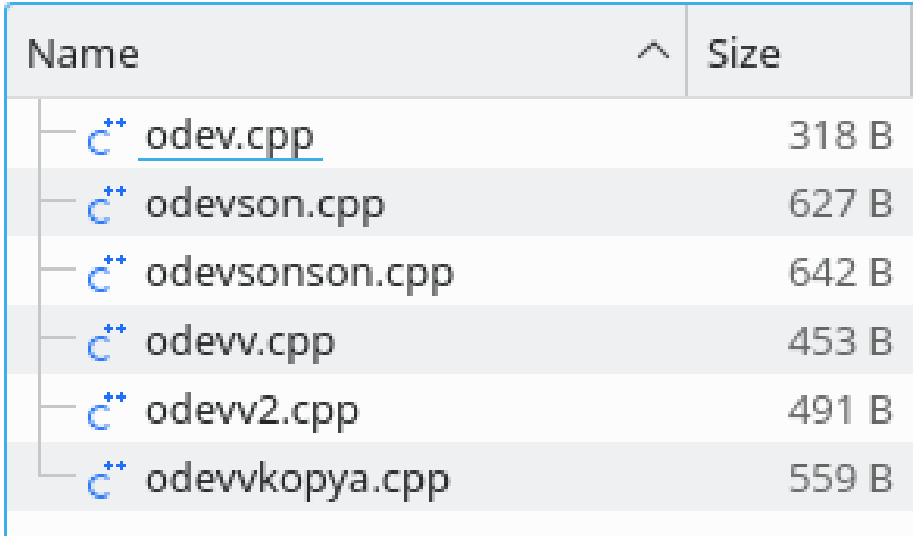


Google Developer Student Clubs

Çankaya University

Çankaya University
Computer Engineering Society

Dur Şu Ödevin Bir Yedeğini Alayım...



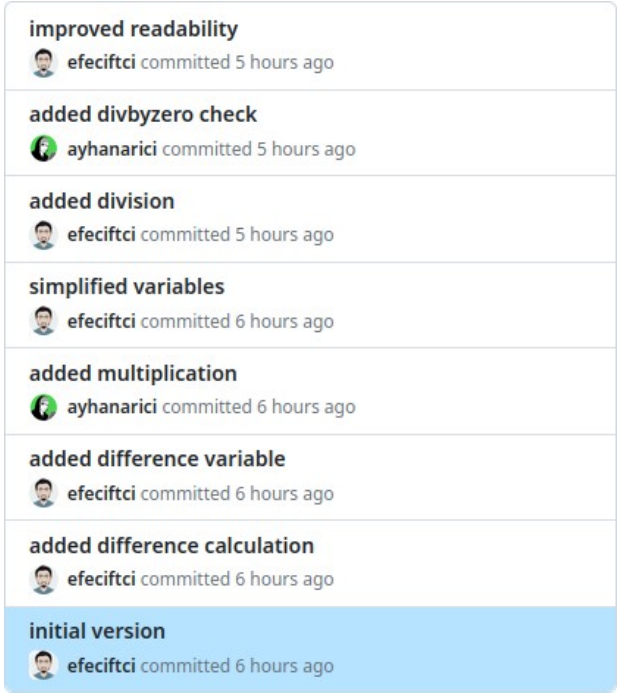
Sürüm Kontrol Sistemleri

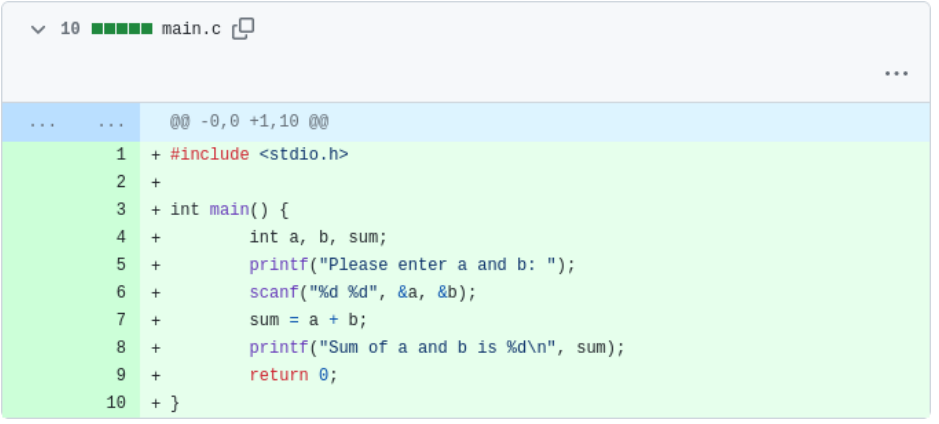
- Sürüm kontrol sistemleri, yazılım geliştirmenin temel bir parçasıdır ve kod ile dökümanlar gibi kaynaklar üzerindeki değişiklikleri yöneten ve kontrol altında tutan yazılımlardır.
- Geliştiriciler, üzerinde çalıştıkları projelerdeki dosyaların farklı anlardaki durumlarını sürüm kontrol sistemleri üzerinde kayıt altına alabilirler.
- Bu kayıtlara “revizyon” denir.
- Bu sistem sayesinde herhangi bir anda eski revizyonlara geri dönüp bakılabilir, proje ve projeyi oluşturan dosyaların farklı revizyonları arasında karşılaştırma yapılabilir, veya proje eski bir revizyona geri döndürülebilir.

Sürüm Kontrol Sistemleri

- Aynı proje üzerinde birden fazla kişi çalışabilir.
- Sürüm kontrol sistemleri, yazılım geliştirme projelerini işbirliği yapılabilir ve yönetilebilir kılar.
- Geliştiricilerin eş zamanlı olarak aynı projede çalışabilmesini, projenin geçmiş durumlarına erişebilmesini ve değişiklikleri takip edebilmesini sağlar. Bu da yazılım geliştirme süreçlerini daha etkili ve düzenli hale getirir.
- Revision Control System (1982), Concurrent Versions System (1986), Visual SourceSafe (1994), Subversion (2000), Bitkeeper (2000), Git (2005), Mercurial (2005) ...

Örnek: Hesap Makinesi





Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

2 main.c

```
@@ -6,5 +6,7 @@ int main() {
6  6          scanf("%d %d", &a, &b);
7  7          sum = a + b;
8  8          printf("Sum of a and b is %d\n", sum);
   9  +       sum = a - b;
   10 +       printf("Difference between a and b is %d\n", sum);
9  11         return 0;
10 12     }
```

Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

6 main.c

```
...  ...   @@ -1,12 +1,12 @@
 1    1    #include <stdio.h>
 2    2
 3    3    int main() {
 4         -      int a, b, sum;
      4    +      int a, b, sum, difference;
 5    5          printf("Please enter a and b: ");
 6    6          scanf("%d %d", &a, &b);
 7    7          sum = a + b;
 8    8          printf("Sum of a and b is %d\n", sum);
 9         -      sum = a - b;
10         -      printf("Difference between a and b is %d\n", sum);
      9    +      difference = a - b;
     10    +      printf("Difference between a and b is %d\n", difference);
11   11          return 0;
12   12    }
```

Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

4 main.c

```
@@ -1,12 +1,14 @@
#include <stdio.h>

int main() {
-       int a, b, sum, difference;
+       int a, b, sum, difference, multiplication;
        printf("Please enter a and b: ");
        scanf("%d %d", &a, &b);
        sum = a + b;
        printf("Sum of a and b is %d\n", sum);
        difference = a - b;
        printf("Difference between a and b is %d\n", difference);
+       multiplication=a*b;
+       printf("Multiplication of a and b is %d\n", multiplication);
        return 0;
}
```

Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

14 main.c

```
@@ -1,14 +1,14 @@
#include <stdio.h>

int main() {
-       int a, b, sum, difference, multiplication;
+       int a, b, result;
        printf("Please enter a and b: ");
        scanf("%d %d", &a, &b);
-       sum = a + b;
-       printf("Sum of a and b is %d\n", sum);
-       difference = a - b;
-       printf("Difference between a and b is %d\n", difference);
-       multiplication=a*b;
-       printf("Multiplication of a and b is %d\n", multiplication);
+       result = a + b;
+       printf("Sum of a and b is %d\n", result);
+       result = a - b;
+       printf("Difference between a and b is %d\n", result);
+       result=a*b;
+       printf("Multiplication of a and b is %d\n", result);
        return 0;
}
```

Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

5 main.c

```
@@ -2,13 +2,16 @@
 
 int main() {
         int a, b, result;
+        float divresult;
         printf("Please enter a and b: ");
         scanf("%d %d", &a, &b);
         result = a + b;
         printf("Sum of a and b is %d\n", result);
         result = a - b;
         printf("Difference between a and b is %d\n", result);
-        result=a*b;
+        result = a * b;
         printf("Multiplication of a and b is %d\n", result);
+        divresult = (float) a / b;
+        printf("Division of a by b is %f\n", divresult);
         return 0;
 }
```

Örnek: Hesap Makinesi

improved readability
efeciftci committed 5 hours ago

added divbyzero check
ayhanarici committed 5 hours ago

added division
efeciftci committed 5 hours ago

simplified variables
efeciftci committed 6 hours ago

added multiplication
ayhanarici committed 6 hours ago

added difference variable
efeciftci committed 6 hours ago

added difference calculation
efeciftci committed 6 hours ago

initial version
efeciftci committed 6 hours ago

8 main.c

```
@@ -11,7 +11,11 @@ int main() {
11  11          printf("Difference between a and b is %d\n", result);
12  12          result = a * b;
13  13          printf("Multiplication of a and b is %d\n", result);
14      -       divresult = (float) a / b;
15      -       printf("Division of a by b is %f\n", divresult);
        14  +   if(b==0)
        15  +           printf("You cannot divide by zero!");
        16  +   else {
        17  +           divresult = (float) a / b;
        18  +           printf("Division of a by b is %f\n", divresult);
        19  +   }
16  20          return 0;
17  21  }
```

Git Tarihçesi

- 2005 yılı öncesinde Linux işletim sistemi çekirdeği, kapalı kaynak kodlu olan BitKeeper sistemi üzerinde geliştirilmekteydi.
- BitKeeper'ın sahibi olan firmanın, Linux geliştiricilerine tanımış olduğu ücretsiz kullanım iznini iptal etmesi üzerine Linus Torvalds ve diğer çekirdek geliştiricileri kendi sürüm kontrol sistemlerini geliştirmeye başlarlar.
- Bu yeni sürüm kontrol sistemi Git adını alır.

Git Kullanımı

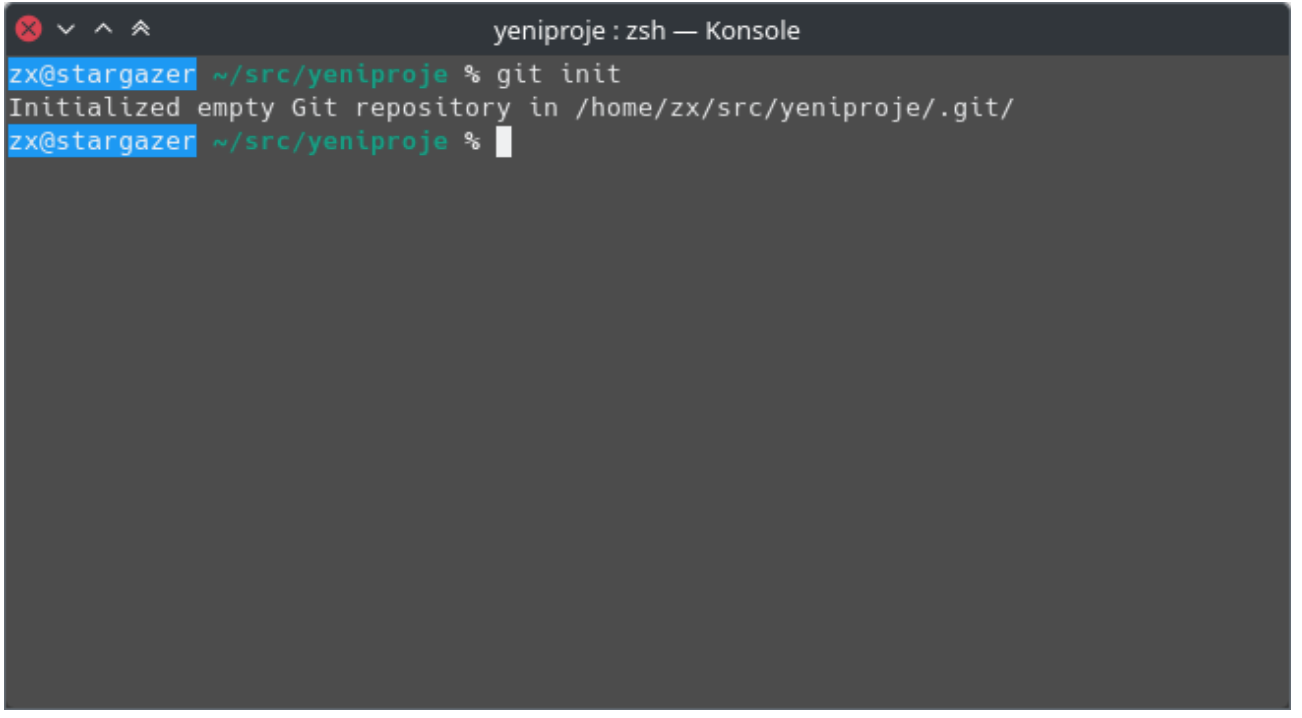
- Projelerinizi Git ile yönetmeye başlamanın birden fazla yolu var:
  - Bilgisayardaki proje dizininde yerel bir Git deposu oluşturabilirsiniz,
  - Şirket gibi ortamlarda tüm çalışanların ortak erişebileceği bir Git sunucusu kurabilirsiniz,
  - GitHub gibi dünyaya açık hizmetlerden faydalanabilirsiniz.

  https://git-scm.com/downloads

- Git'i hem komut satırından, hem görsel uygulamalardan kullanmak mümkündür.

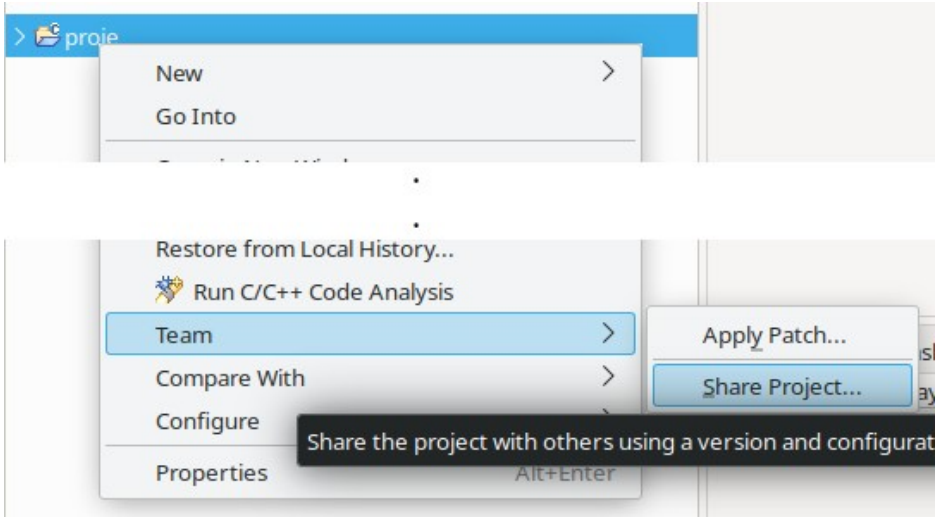
Yeni Git Deposu Oluşturma

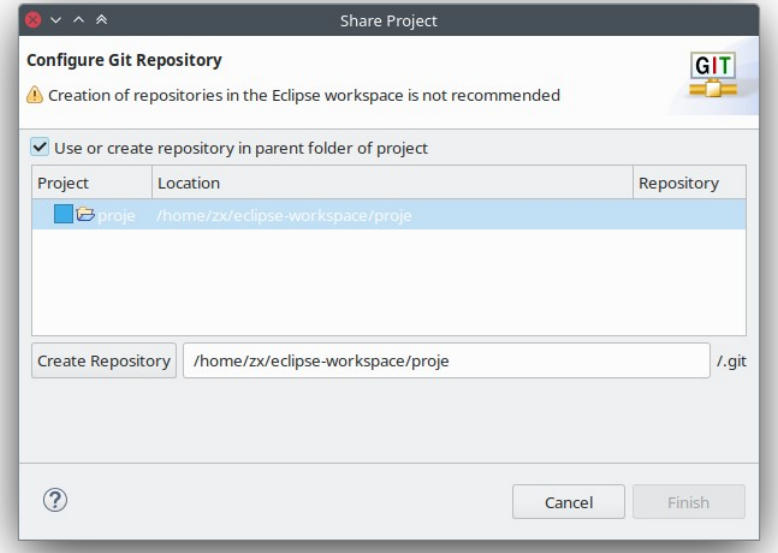
- `git init` komutu ile proje dizininizde yeni bir Git deposu oluşturabilirsiniz.



Yeni Git Deposu Oluşturma

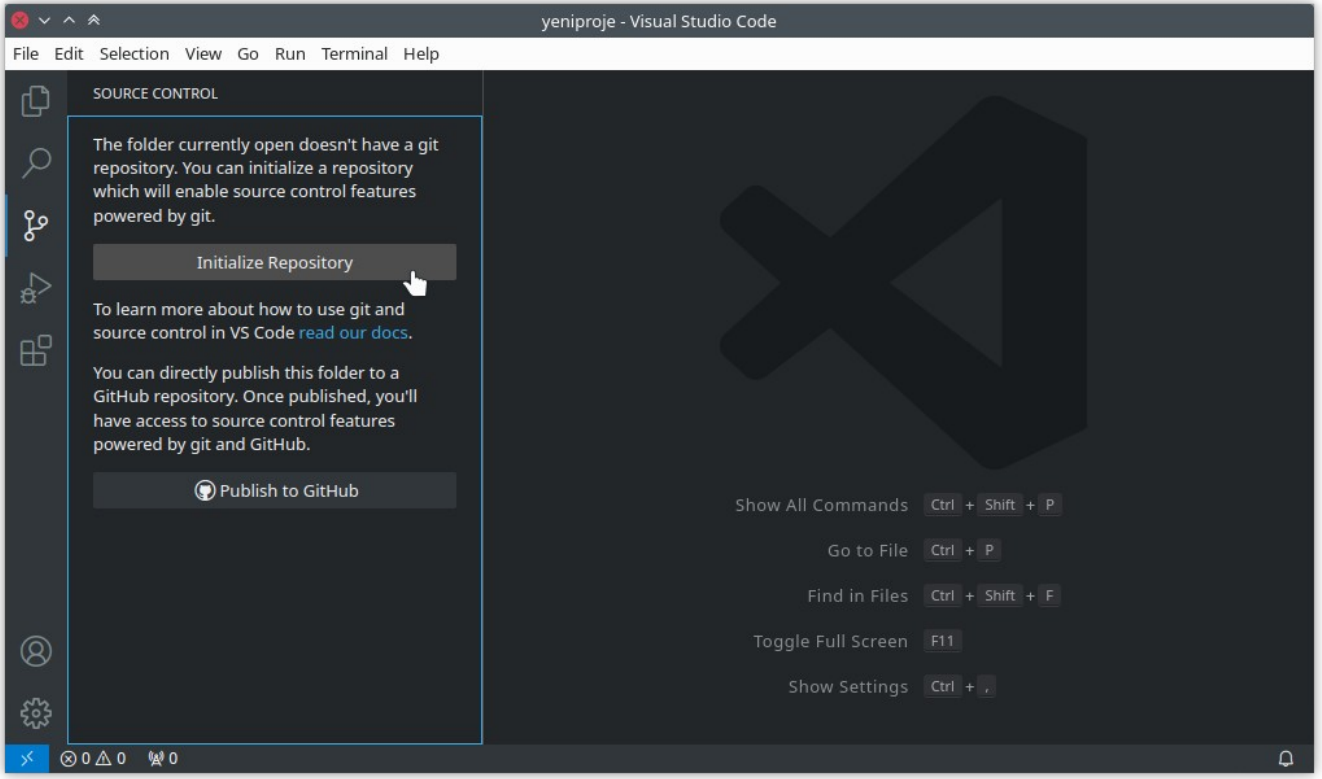
- Grafik arayüz ile de mümkün; örneğin Eclipse:





Yeni Git Deposu Oluşturma

- Grafik arayüz ile de mümkün; örneğin Visual Studio Code:

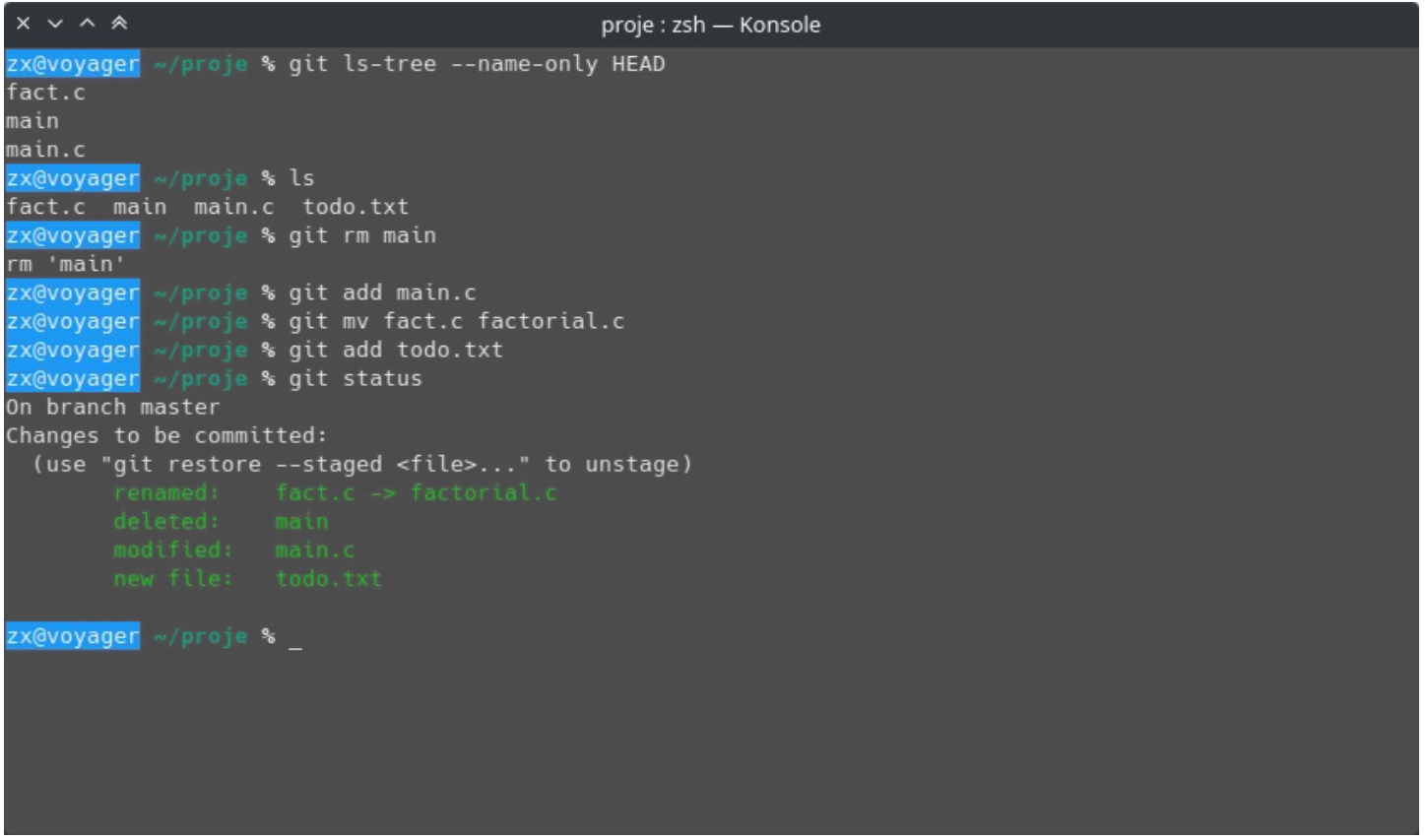


Dosya Ekleme, Taşıma, Silme

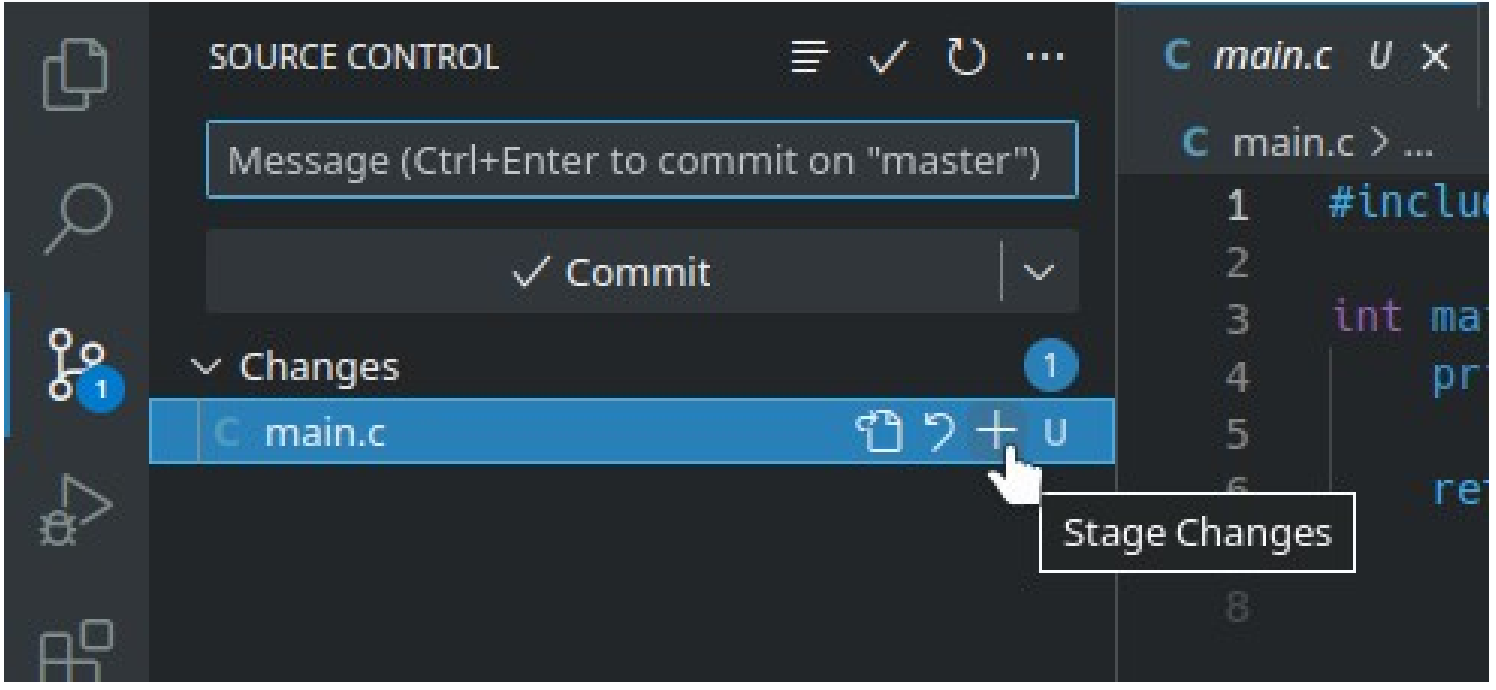
- Proje ağacındaki bir sonraki revizyona;
  - Yeni dosya eklemek veya mevcut dosyalarda yapılan değişiklikleri tanıtmak için `git add` komutu,
  - Bir dosyayı başka bir dizine taşımak veya adını değiştirmek için `git mv` komutu,
  - Bir dosyayı proje ağacından çıkartmak için `git rm` komutu

  kullanılır.

Dosya Ekleme, Taşıma, Silme



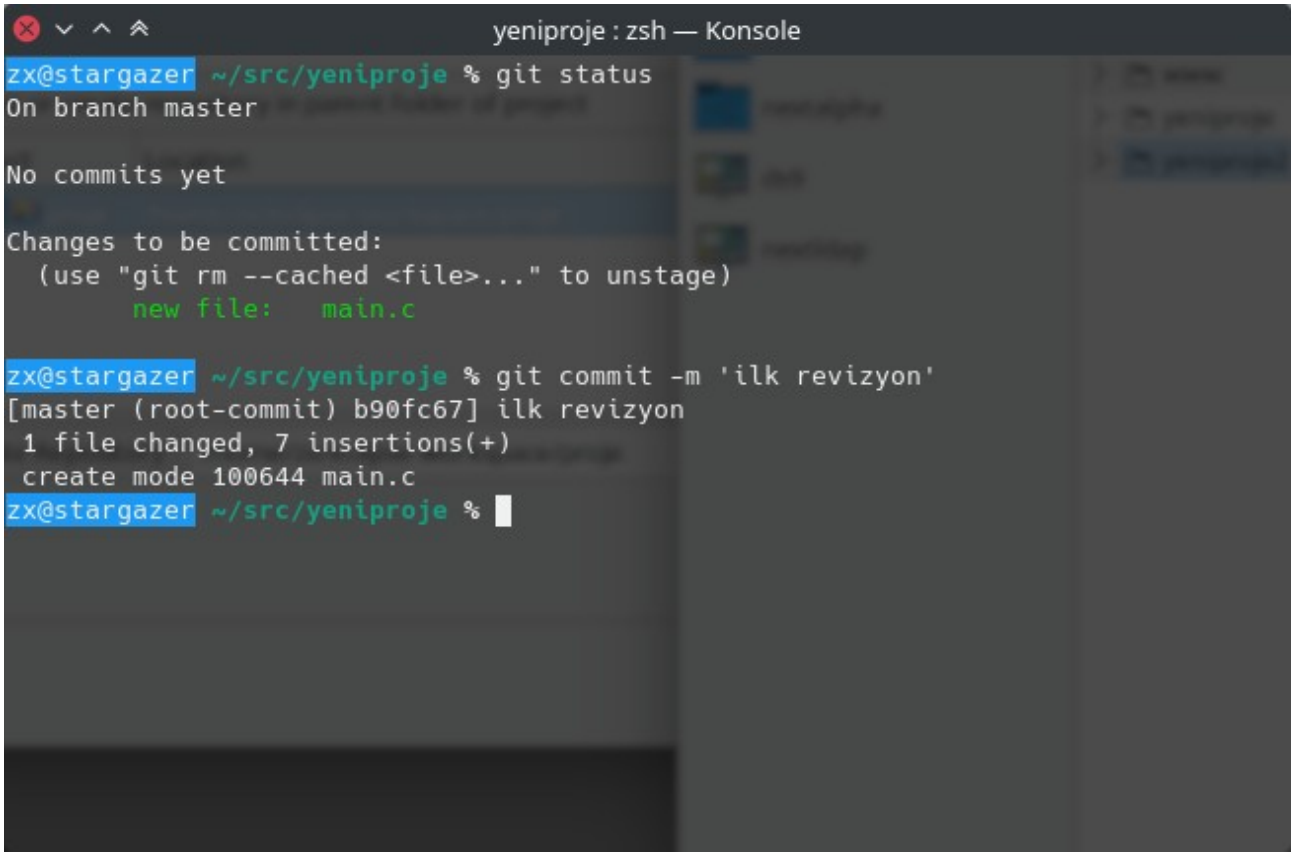
Dosya Ekleme, Taşıma, Silme



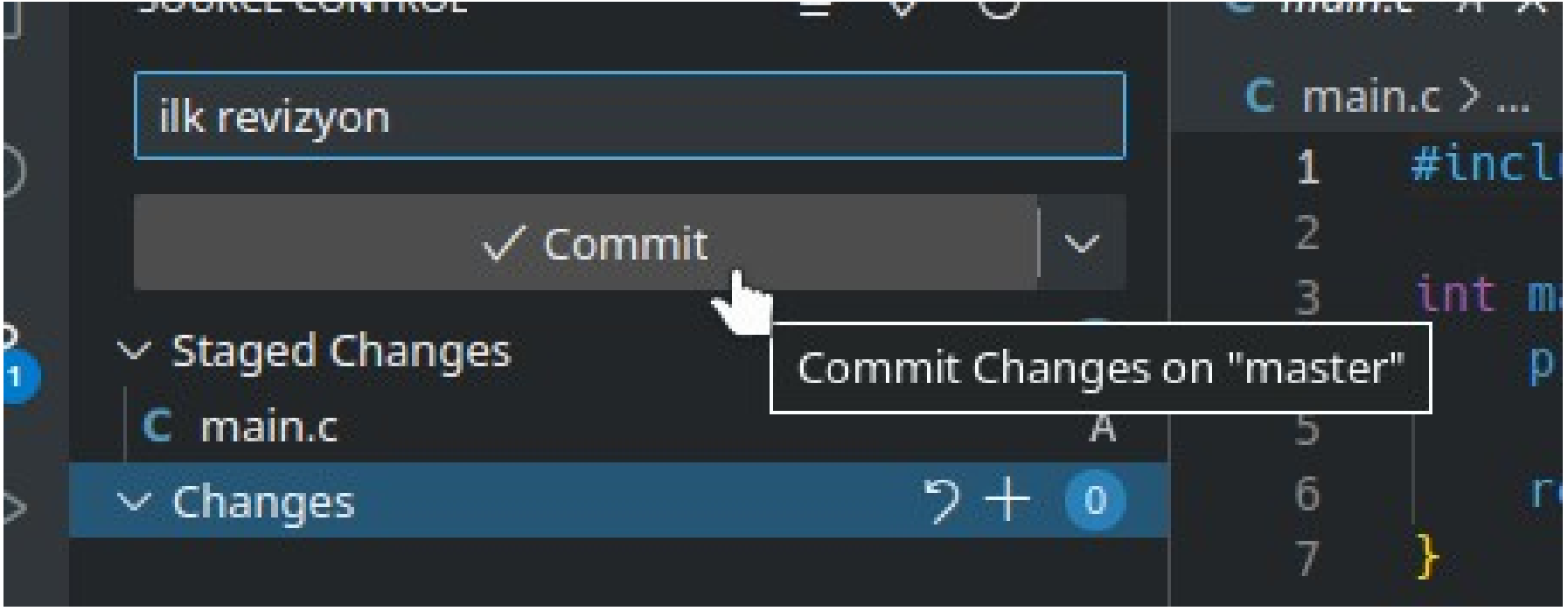
Revizyon Oluşturma

- Yeni revizyon için tüm ekleme/çıkartma/taşıma işlemleri bittiğinde `git commit` komutu ile revizyon oluşturulur.
- Her revizyonun bir açıklaması olması gereklidir.
  - `git commit` bu açıklamayı harici bir metin editöründen okur,
  - `git commit -m MESAJ` kısa açıklamaların hızlıca komut satırından girilmesini sağlar.

Revizyon Oluşturma

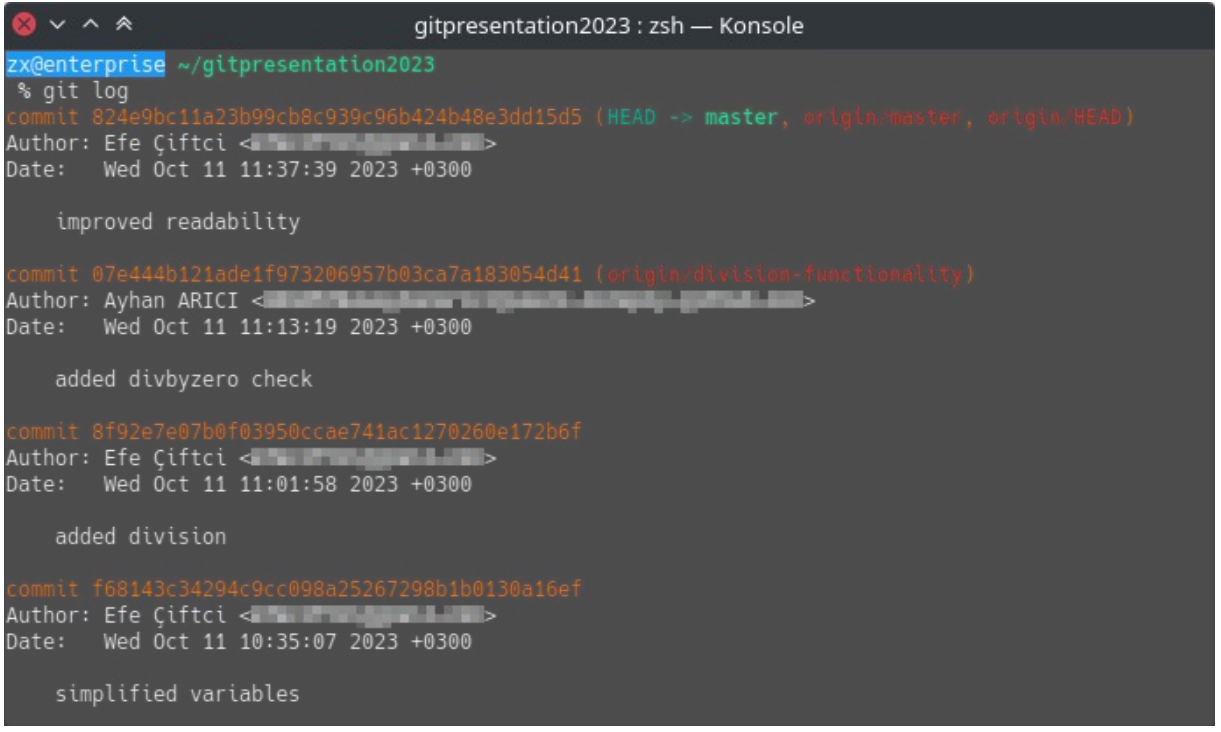


Revizyon Oluşturma



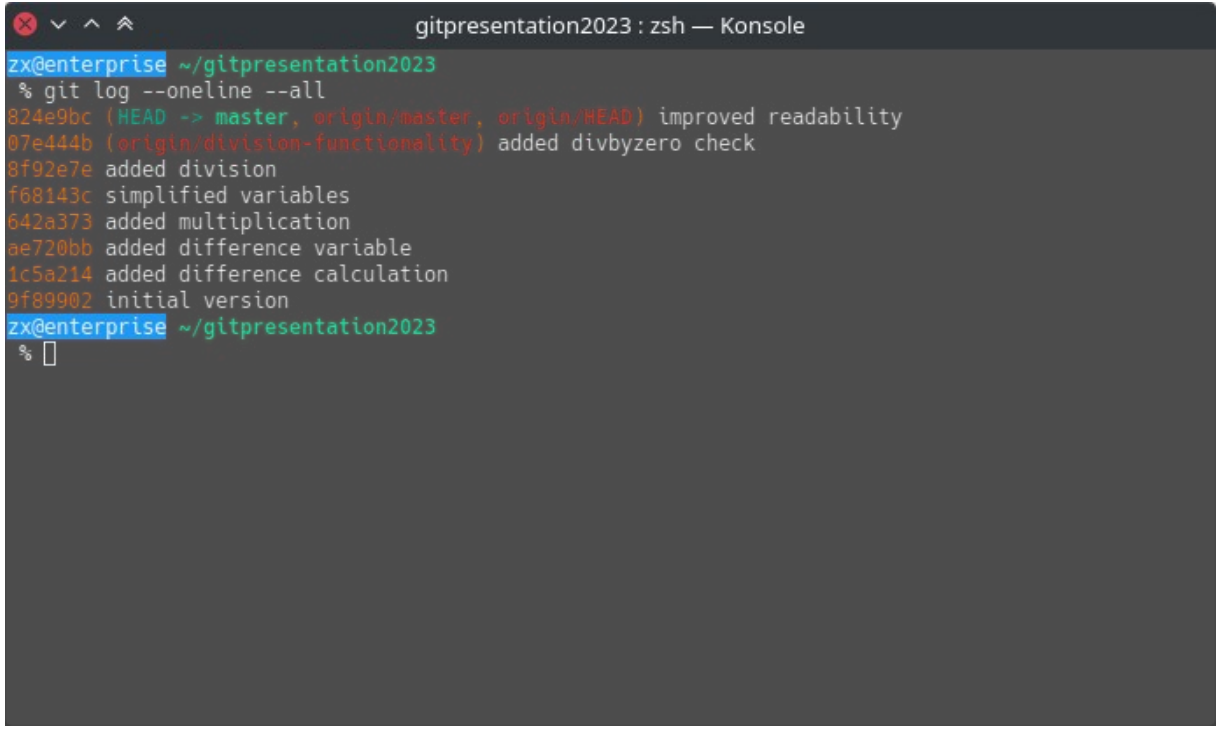
Revizyonlar Arası Karşılaştırma

- `git log` komutu ile projeye ait tüm revizyonlar listelenebilir:



Revizyonlar Arası Karşılaştırma

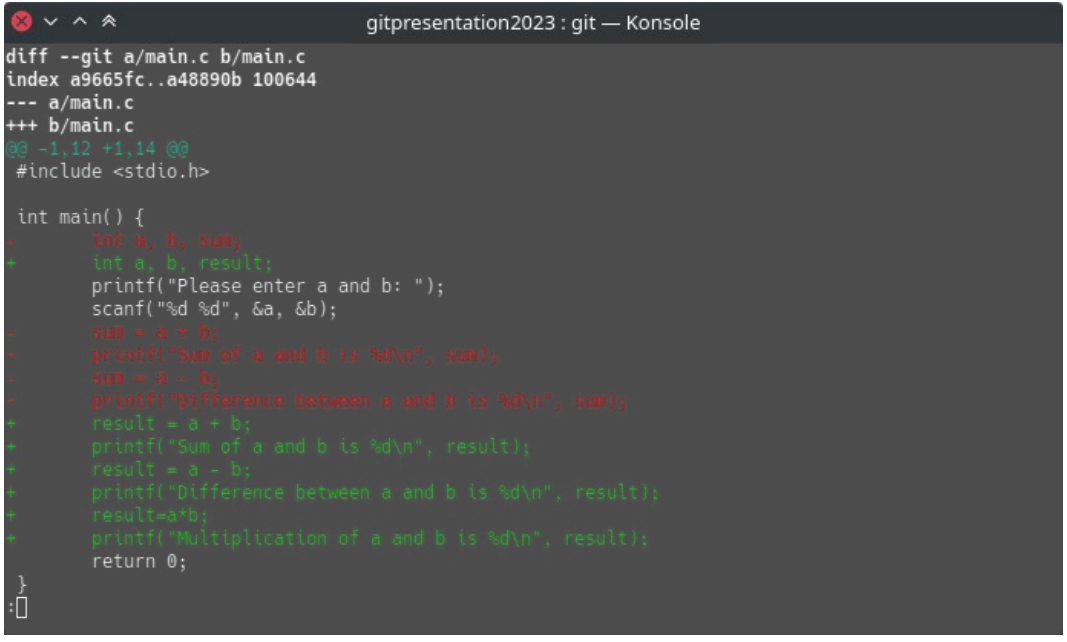
- `git log --oneline` komutu ile projeye ait tüm revizyonlar özet olarak listelenebilir:



Revizyonlar Arası Karşılaştırma

- `git diff REV1 REV2` komutu ile revizyon numarası verilen iki revizyon arasındaki farklılıklar listelenebilir.

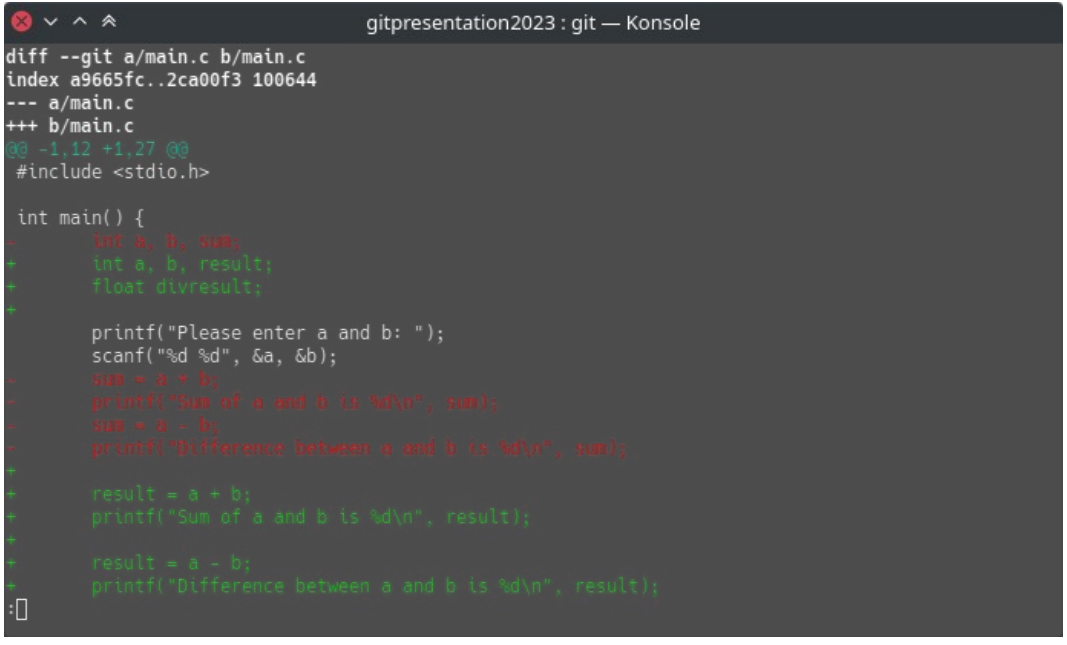
```
git diff 1c5a214 f68143c
```



Revizyonlar Arası Karşılaştırma

- Birden fazla dosyanın değiştiği revizyonlarda sadece belli dosyalardaki farklıkları görüntülemek için `git diff REV1 REV2 DOSYA` komutu kullanılabilir.

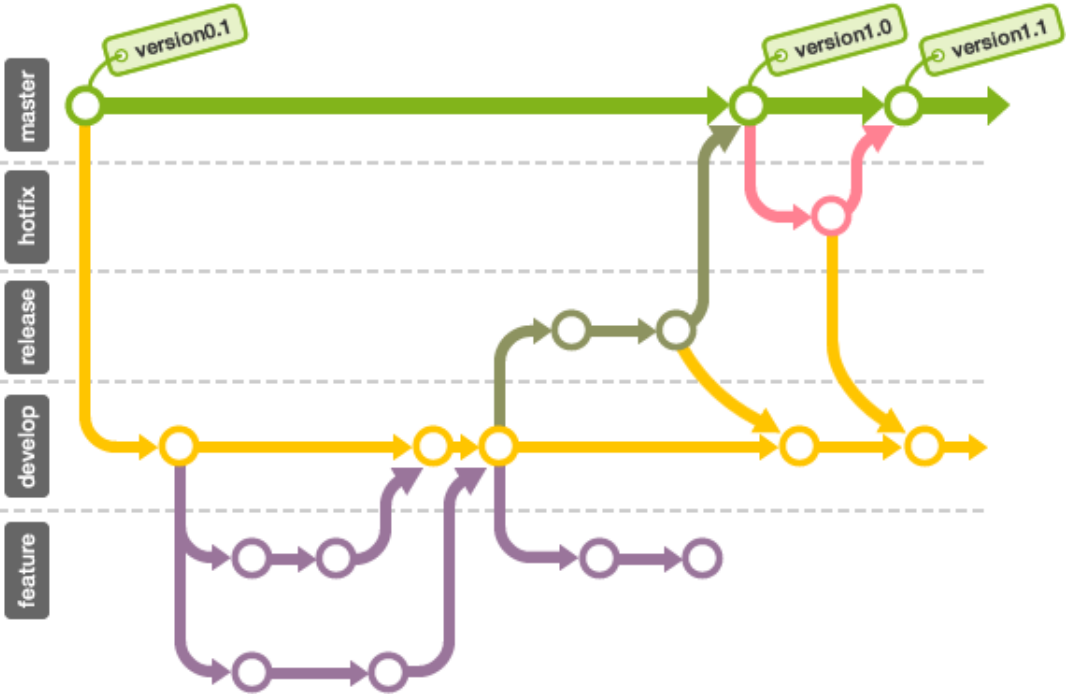
```
git diff 1c5a214 824e9bc main.c
```



Branch Kavramı

- Git ile yönetilen projeler üzerinde farklı dallar (branch) oluşturulabilir.
- Bu dallar, projenin farklı özelliklerini veya hata düzeltmelerini geliştirmek için kullanılır.
- Her dal, projenin bağımsız bir kopyasını temsil eder ve geliştiriciler farklı görevler üzerinde çalışırken ana projeyi etkilemeden değişiklik yapabilirler.
- İş tamamlandığında veya bir dalın geliştirmesi sona erdiğinde bu dallar ana projeye birleştirilebilir.

Branch Kavramı



Kaynak: Simple Git tutorial for beginners

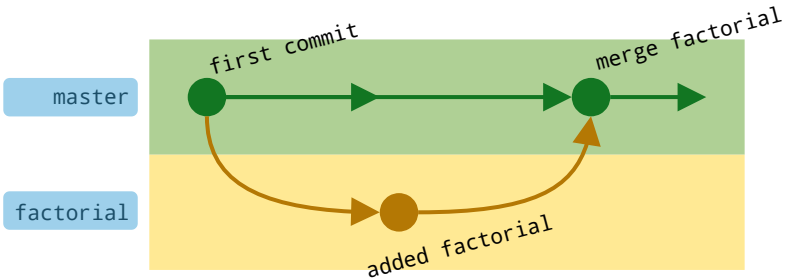
Branch Kavramı

- `git branch` ile mevcut dallar görüntülenir.
- `git branch İSİM` ile yeni bir dal oluşturulur.
- `git checkout İSİM` ile dallar arasında geçiş yapılır.
- `git checkout -b İSİM` ile yeni bir dal oluşturulur ve bu yeni dala geçiş yapılır.

- `git merge İSİM`, İSİM dalını mevcut dal ile birleştirir.
- `git merge --no-ff -m MESAJ İSİM`, İSİM dalını mevcut dal ile MESAJ açıklamalı bir revizyon oluşturarak birleştirir.

Branch Kavramı

```
$ git init
Initialized empty Git repository in /home/zx/proje/.git/
$ git add main.c
$ git commit -m 'first commit'
[master (root-commit) 9ec86ad] first commit
1 file changed, 4 insertions(+)
create mode 100644 main.c
$ git checkout -b factorial
Switched to a new branch 'factorial'
$ git add main.c fact.c
$ git commit -m 'added factorial'
[factorial 1536825] added factorial
2 files changed, 7 insertions(+)
create mode 100644 fact.c
$ git checkout master
Switched to branch 'master'
$ git merge --no-ff -m 'merge factorial' factorial
Merge made by the 'ort' strategy.
fact.c | 6 ++++++
main.c | 1 +
2 files changed, 7 insertions(+)
create mode 100644 fact.c
$ git log --oneline --graph
*   36fdcd4 (HEAD -> master) merge factorial
|\
| * 1536825 (factorial) added factorial
|/
* 9ec86ad first commit
```



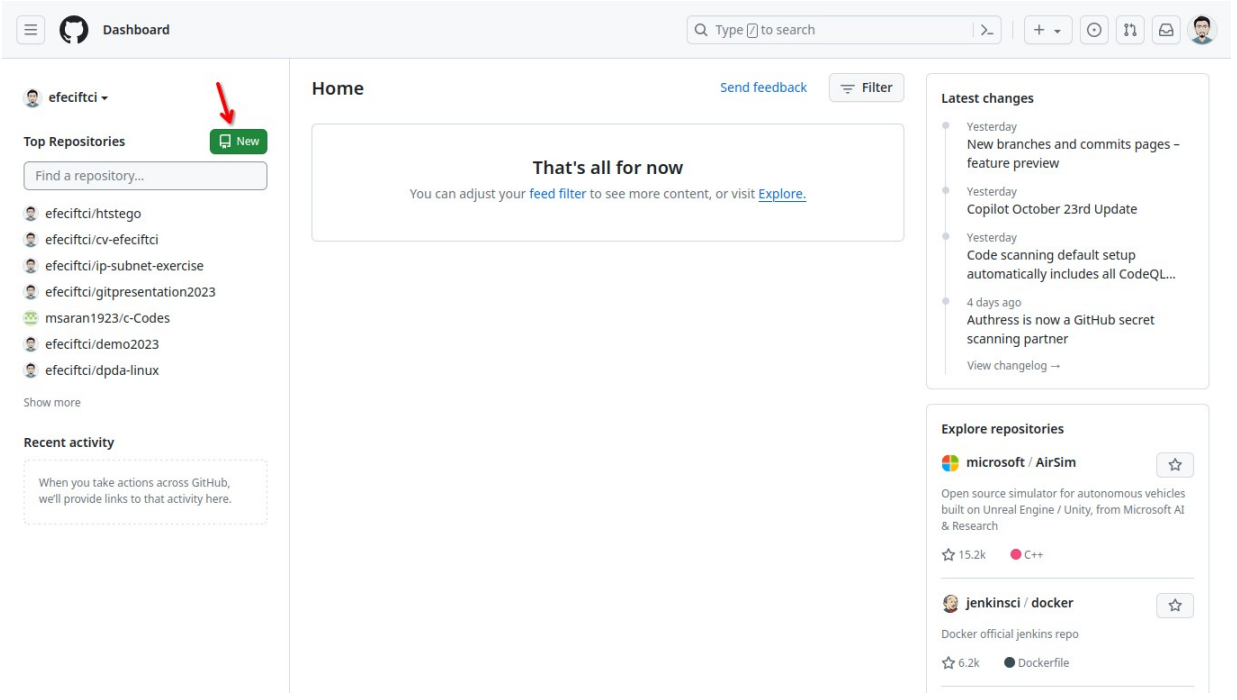
GitHub

- Github, küresel bir Git proje depolama hizmetidir.

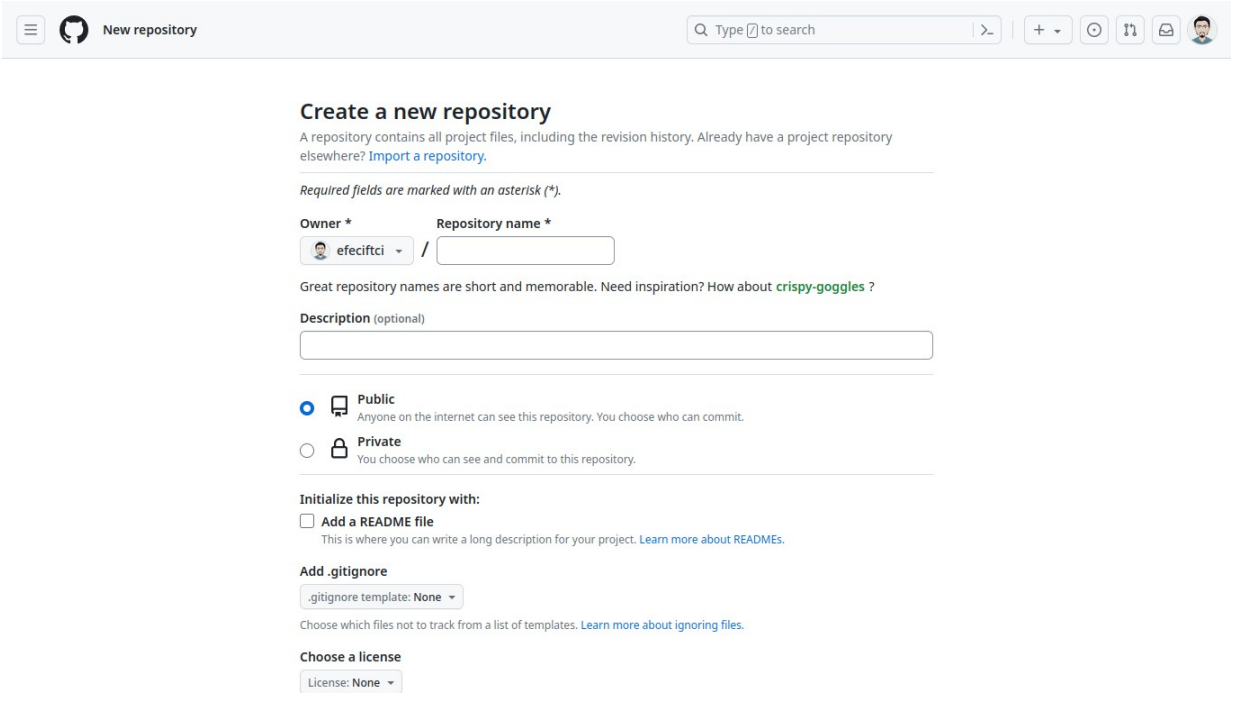
  https://github.com/

- Projelerinizi çevrimiçi olarak depolamanıza ve yönetmenize olanak tanır.
- Projelerinizi diğer geliştiricilerle paylaşabilir ve işbirliği yapabilirsiniz.
- Ücretsiz bir GitHub hesabı oluşturarak projelerinizi oluşturabilir ve bunların görünürlüğünü dilediğiniz gibi ayarlayabilirsiniz.
- Diğer geliştiricilere ait projeleri inceleyebilir, bu projelerin kopyalarını oluşturabilir ve hata düzeltmeleri veya yeni özellikler ekleyebilirsiniz.

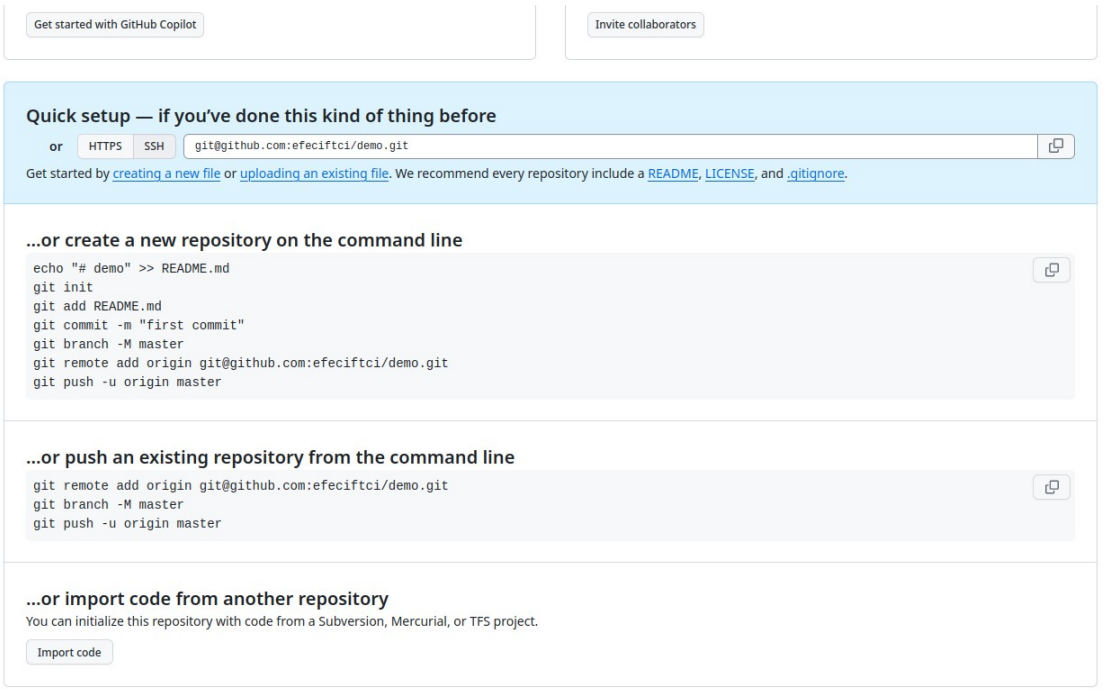
GitHub Üzerinde Proje Yönetimi



GitHub Üzerinde Proje Yönetimi

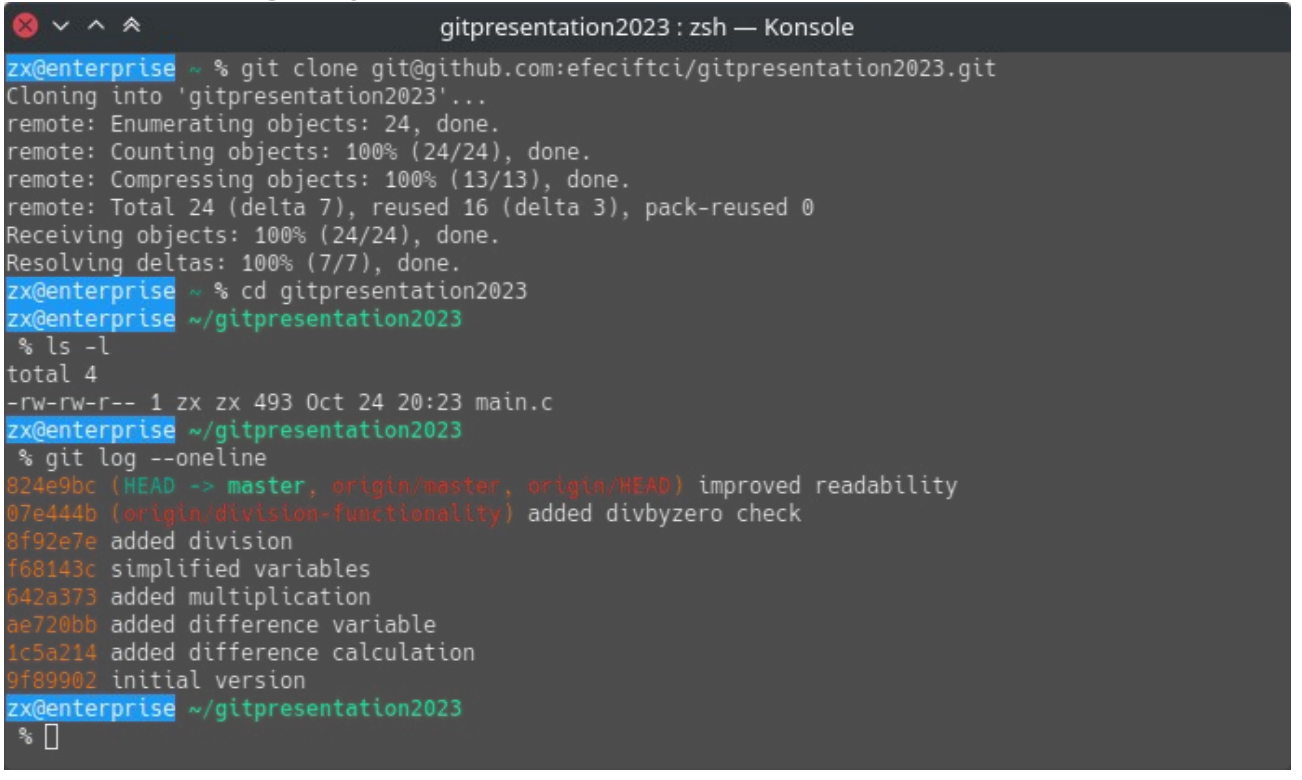


GitHub Üzerinde Proje Yönetimi



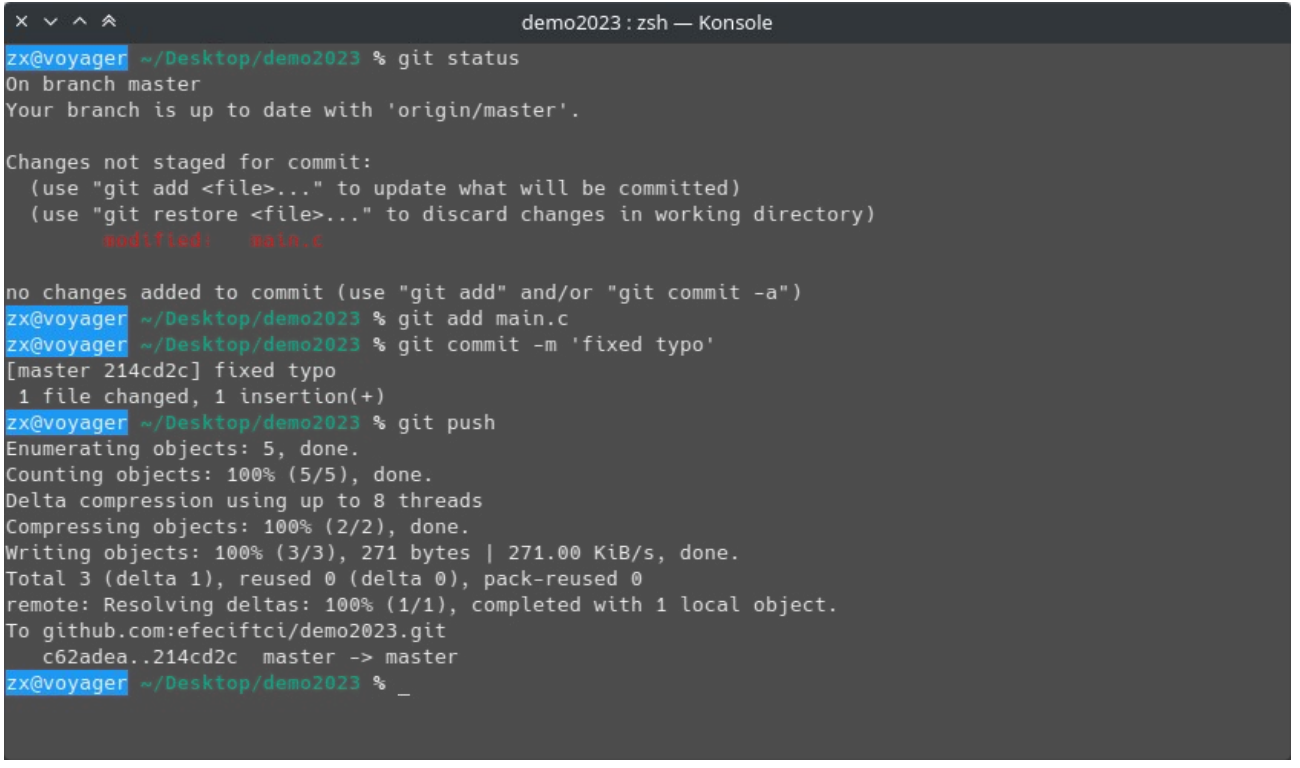
GitHub Üzerinde Proje Yönetimi

- Var olan bir proje `git clone UZAK_REPO_ADRESİ` komutu ile projenin tüm geçmişi ile birlikte bilgisayara indirilebilir.



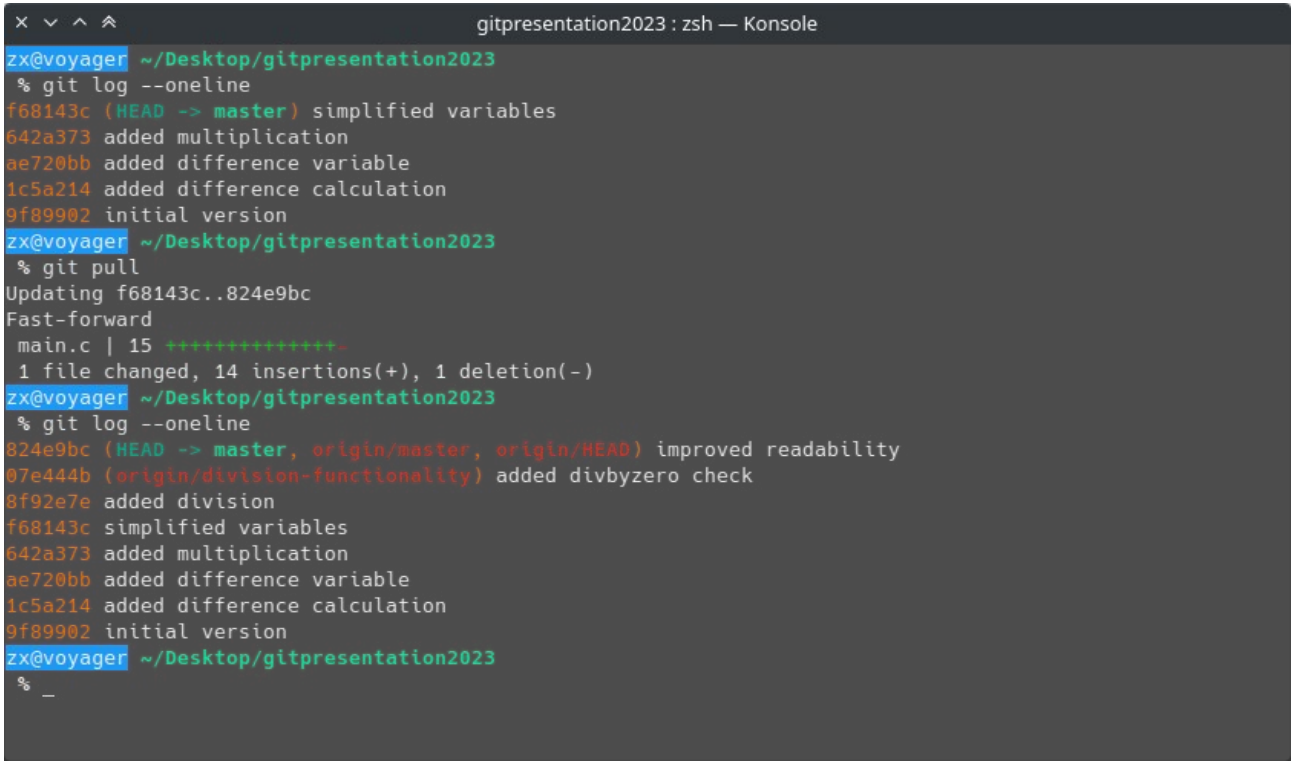
GitHub Üzerinde Proje Yönetimi

- Kendi bilgisayarınızda oluşturduğunuz revizyonlar karşı sunucuya `git push` komutu ile iletilir.



GitHub Üzerinde Proje Yönetimi

- Bir projenin bilgisayarda tutulan kopyasını, uzak sunucudaki güncel revizyona çekmek için `git pull` komutu kullanılır.



GitHub Üzerinde Proje Yönetimi

- Projelerin başarılı bir şekilde yönetilmesi ve kullanıcıların projeyi anlamaları için dökümantasyon önemlidir.
- GitHub, Markdown dilini kullanarak proje dökümantasyonunuzu kolayca oluşturmanızı sağlar.
  - README.md
  - Changelog.md
  - License.md
  - ...

GitHub Üzerinde Proje Yönetimi

```
# Git ve GitHub'a Giriş
Çankaya Üniversitesi Bilgisayar Mühendisliği Bölümü üyesi Öğr. Gör. Dr. Efe
Çiftci tarafından verilecek olan bu seminerde katılımcılarla aşağıdaki
konular hakkında bilgiler paylaşılacaktır:
- Sürüm kontrol sistemleri
- Git tarihçesi
- En sık kullanılan Git özellikleri
  - Depo oluşturma
  - Revizyon oluşturma
  - Branch oluşturma ve birleştirme
  - ve daha fazlası!

![poster](assets/poster.png)

**Tarih:** 31 Ekim 2023
**Saat:** 13:30
**Yer:** Merkez Kampüs Mavi Salon

## Slayt Notları
Sunum sırasında kullanılacak olan slaytlara [konuşmacı kişisel web
sayfasından](https://efeciftci.com/) erişebilirsiniz.
```

Git ve GitHub'a Giriş

Çankaya Üniversitesi Bilgisayar Mühendisliği Bölümü üyesi Öğr. Gör. Dr. Efe Çiftci tarafından verilecek olan bu seminerde katılımcılarla aşağıdaki konular hakkında bilgiler paylaşılacaktır:

- Sürüm kontrol sistemleri
- Git tarihçesi
- En sık kullanılan Git özellikleri
  - Depo oluşturma
  - Revizyon oluşturma
  - Branch oluşturma ve birleştirme
  - ve daha fazlası!

Tarih: 31 Ekim 2023
Saat: 13:30
Yer: Merkez Kampüs Mavi Salon

Slayt Notları

Sunum sırasında kullanılacak olan slaytlara konuşmacı kişisel web sayfasından erişebilirsiniz.

Sorularınız?

Dinlediğiniz İçin Teşekkürler!

Bu sunum özgür işletim sistemi KDE neon üzerinde özgür ofis yazılımı LibreOffice ile üretilmiştir.

"Free as in free speech, not free beer"